HIGH- LEVEL TEST STRATEGY

(E-commerce checkout process)

	Name	Role	Signature	Date
Prepared By:	Mary Francis	Intermediate QA		January 1, 2024
Reviewed By:				
Approved By:				

TABLE OF CONTENT

1. Introduction

This test strategy is provided to assist the QA team and provide a guide to govern the strategic approach of the test effort; it defines the general approach to be employed when testing the software and when evaluating the results of that testing. Every new QA employed to the company is expected to go through this documentation to understand what is required to effectively test any software. This documentation feeds the Test plan of any software product to be prepared.

1.1 Review

This document will be reviewed at least twice yearly to accommodate new innovation made in software engineering. All new changes made must be effected in all test plans prepared from the test strategy

1.2 Testing Motivators

All testing efforts employed during any project must :

1. Adhere to Government standard and mandate
2. Satifies User Acceptance Criteria
3. Ensure that project meets Owners requirement
4. Conform to industry rules and standard

2. Testing Approach

This is a high-level test strategy for an e-commerce checkout process. This is provided to assist the QA team and provide a guide to govern the strategic approach of the test effort; it defines the general approach to be employed when testing the checkout process and the results of that testing

2.1 Test Types

2.1.1 Unit Test

Unit testing is the initial testing of new and/or changed code in the system. The purpose of unit testing is to allow the developer to confirm the functionality provided by a single unit or component of code. Additionally, wherein one component cannot function without interacting with another component, the test shall include limited interactions.

Unit testing shall consist of the following:

1. **Test Scope**:
    - Static testing – Conducting “walkthroughs” and reviews of the design and coded components.
    - Basic path testing – Executing path testing based on normal flow.
    - Condition/multi-condition testing – Executing path testing based on decision points.
    - Data flow testing – Examining the assignment and use of variables in a program.
    - Loop testing – Checking the validity of loop constructs.
    - Error testing – Executing unexpected error conditions.

1. **Test Environment:** Development environment
2. **Test Data:** Manual data created by Developer
3. **Role:** Developer
4. **Deliverables**:

2.1.2 Functional Test

1. **Test Scope:**

Functional testing is a type of software testing that focuses on verifying that a software application behaves according to its specifications and meets the intended functional requirements. The primary goal of functional testing is to ensure that the software application operates as expected, performs the functions it is designed for, and produces accurate results.

2. **Test Environment:** Test environment
3. **Test Data:** Manual data created by Test team
4. **Role:** QA team
5. **Deliverables:** Bug Report and Test report

2.1.3 User Acceptance Test (UAT)

1. **Test Scope**:

The purpose of user acceptance testing (UAT) is to simulate the business environment and emphasize security, documentation, and regression tests. UAT may be performed by a third party (*i.e.,* TCO) in cases where the general user community is large and may provide different goals and objectives for acceptance testing requirements.

UAT shall be conducted to gain acceptance of all functionality from the user community. UAT shall verify that the system meets user requirements as specified.

2. **Test Environment:** Pre-Prod or Implementation
3. **Test Data:** Real Life Data
4. **Role:** Process or Business Users
5. **Deliverables:** Feedback Form

2.1.4 Load Test

1. **Test Scope:**

The purpose of load testing is to identify potential performance problems before they occur in production.

Load testing shall be used to test the performance of the application with near-production (or greater) levels of users accessing the application at the same time as specified by the Supplementary Specifications.

2. **Test Environment:** Pre-Prod or Implementation
3. **Test Data:** Nil
4. **Role:** QA Team
5. **Deliverables:** Test Report

2.1.5 Smoke Test

1. **Test Scope:**

The purpose of smoke testing is to identify any potential issues with the production environment setup before users access the system.

Operational readiness testing shall verify that the application move from the acceptance environment to the production environment was successful.

2. **Test Environment:** Production
3. **Test Data:**Nil
4. **Role:** QA Team
5. **Deliverables:** Bug Report or Sign-off for deployment

2.1.6 Automation Testing

1. **Test Scope:**

The purpose of this testing is to automate or repetitive tasks, regression testing, and critical scenarios.

2. **Test Environment:** Staging
3. **Test Data:**Nil
4. **Role:** QA Team
5. **Deliverables:** Bug Report and Test Report

3. Test environment:

The environments below highlight various testing efforts and responsible parties

1. **Staging Environment** - Mimics the production environment closely to validate end-to-end functionality. QA personnel will perform testing on this environment
2. **UAT Environment** - This environment is for personnel not involved in the development phase of. This is to monitor real-world performance. Enables early detection of any issues before a full-scale release
3. **Production Environment** - Full-scale release to the entire user base after successful testing in the staging and limited release environments.

4. Documentation

The following documentation shall be provided by the QA team

1. Test Cases Document
2. Test Strategy Document
3. Test Report
4. Bug Report"

5. Tools

The following Test tools shall be employed:

1. Google sheet for documentation
2. Jira for Task Management
3. Cypress for web automation
4. Slack for Communication